

JANVIER 2024

RAPPORT DE STAGE BTS SIO SLAM

SEMAINE 1/2

PRÉPARÉ PAR NAM LUAN

CHRISTIAN LOUBOUTIN

DÉCOUVERTE DE L'ENVIRONNEMENT



Avant de commencer à travailler j'ai découvert les différentes équipes présentent au sein de l'entreprise ainsi que les différents outils utilisé. Pour ce faire j'ai effectuer des appels avec chaque personne à la tête des différentes équipes. Une fois cela fait j'ai eu une meilleur vision sur l'ensemble de mon environnement de travail.



PRÉSENTATION DES DIFFÉRENTS OUTILS



OpCon™ est une Plateforme d'Automatisation Digitale multi-environnements, développée par SMA Technologies, capable de traiter de façon centralisée l'ensemble de l'automatisation d'une exploitation informatique.



Le WALLIX Bastion est une solution modulaire qui permet aux organisations d'adapter son déploiement en fonction de leurs besoins, en ajoutant des fonctionnalités au fur et à mesure, pour mettre en place une politique de sécurité des accès à priviléges durable.



Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS.

APERÇU DES **MISSION**

Durant mes 2 premières semaines, on m'a donné 2 mission :

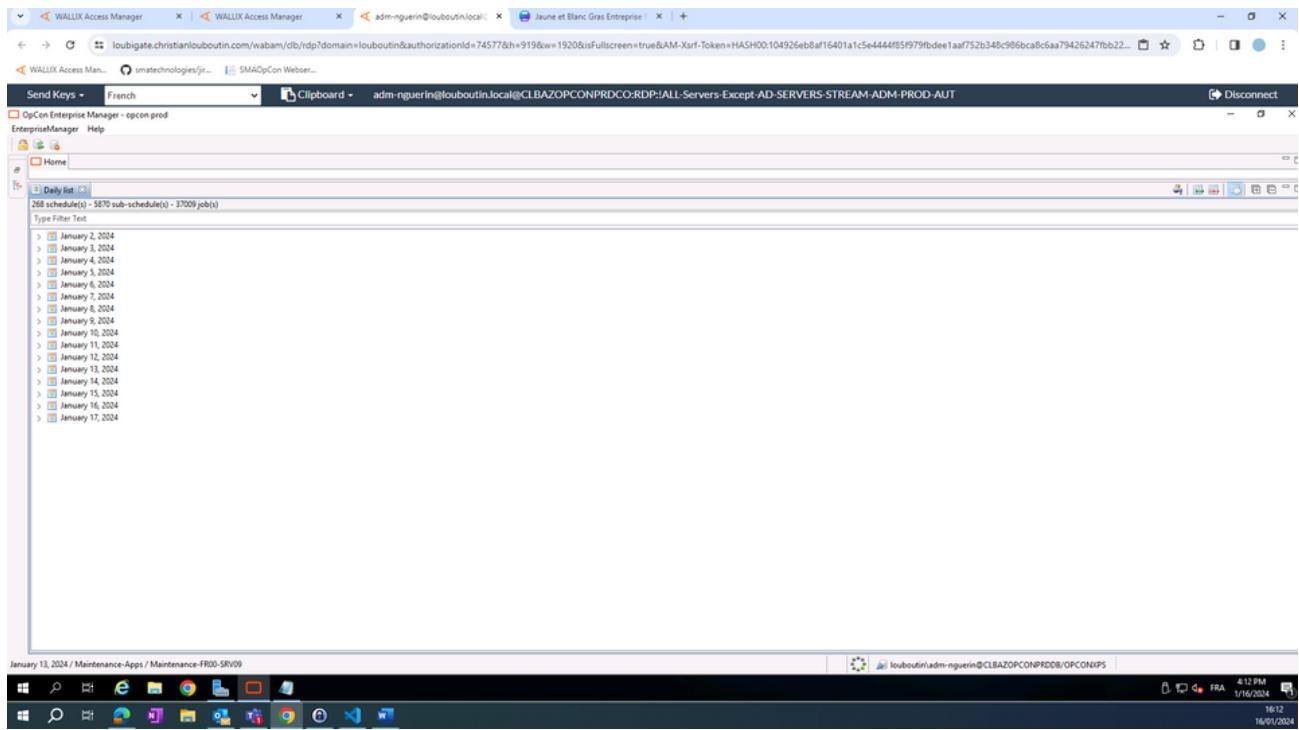
- interconnecter via api, l'ordonnanceur de l'entreprise à l'outil Jira
- Mettre en place le stop/start M3 Homologation. Optimiser les scripts existant



RAPPORT DE STAGE SEMAINE 1

APERÇU DES MISSION

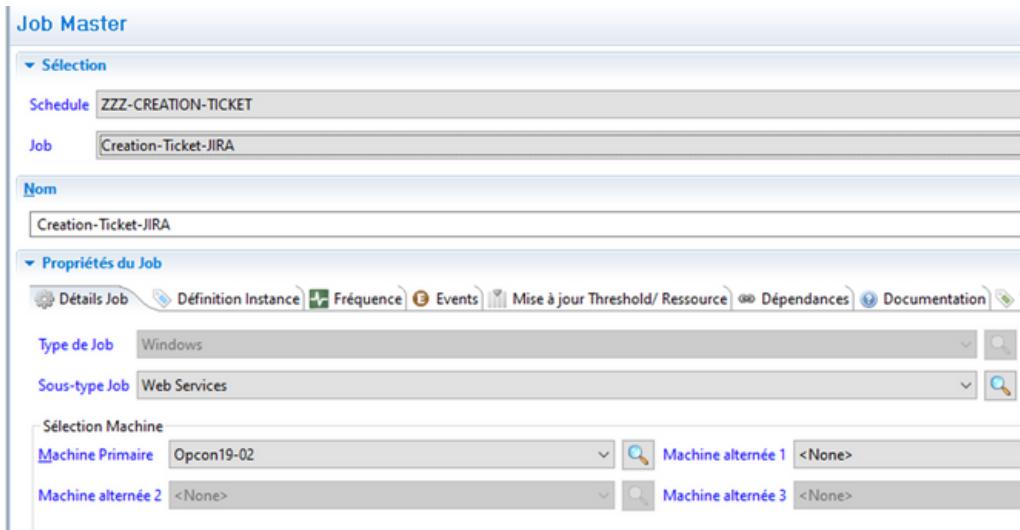
Interface des outils Wallix et Opcon



RAPPORT DE STAGE SEMAINE 1

APERÇU DES MISSION

Création du job dans Opcon permettant d'automatiser la création de ticket Jira



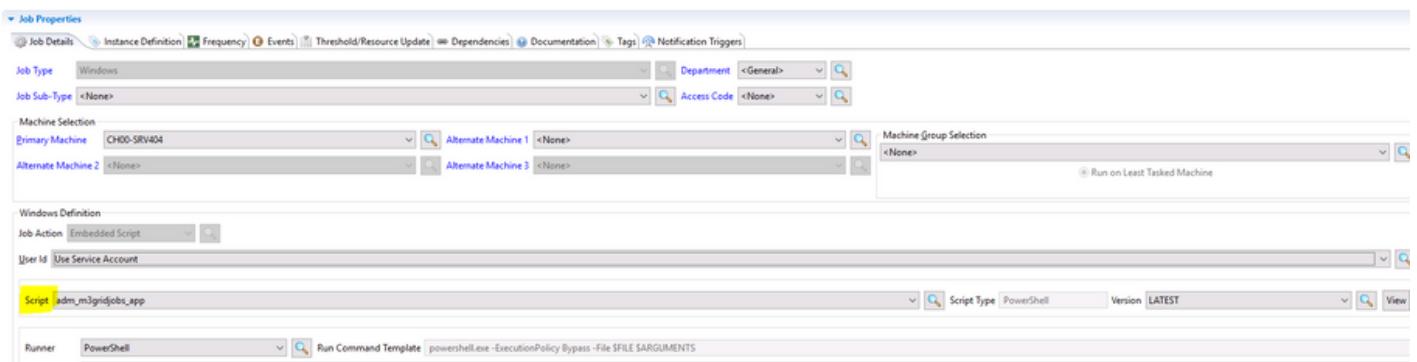
Modification du script du job

```
Body (Json)
File Name
Message Body
{
  "fields": {
    "project": {
      "key": "OT"
    },
    "summary": "Job[[J].Jname] Schedule[[J].Sname] est en erreur",
    "description": {
      "version": 1,
      "type": "doc",
      "content": [
        {
          "type": "paragraph",
          "content": [
            {
              "type": "text",
              "text": "Le job[[J].Jname] du Schedule[[J].Sname] en date du [[$SCHEDULE DATE]] est en erreur avec un code retour = [[J].Cret]] sur le serveur : [[J].Mach]]"
            }
          ]
        }
      ],
      "issuetype": {
        "name": "Incident"
      },
      "priority": {
        "name": "Medium",
        "id": "3"
      }
    }
  }
}
```

RAPPORT DE STAGE SEMAINE 1

APERÇU DES MISSIONS

Exemple d'une page de propriété (Opcon) d'un job contenant un script



Extrait de code que j'ai optimisé
Ce code retourne si le service est arrêté ou démarré et si il a bien effectué cette action

```
14  {
15
16      [Parameter(Mandatory=$false)]
17      [String] $name="",
18
19      [Parameter(Mandatory=$false)]
20      [String] $action=""
21  }
22
23  # Function
24
25  8 references
26  Function ExitWithCode($ExitCode)
27  {
28      $host.SetShouldExit($ExitCode)
29      exit $ExitCode
30  }
31
32  # Initialisation
33
34  $todayDate = Get-Date -Format "yyyy-MM-dd-HHmmssfff"
35
36  # Main
37  Write-host("r----- Windows Service(s) (Stop / start) -----`r")
38  Write-host ("$todaydate Init Job Stop.Start.services from Server $env:COMPUTERNAME with account $env:UserName `r")
39
40  if ($action -eq "Stop") {
41      Write-host ("$todaydate Stop service(s) `r")
42      Get-Service -DisplayName $name | ForEach-Object { if($_.status -eq "running"){Stop-Service $name}}
43      Write-host ("$todaydate service(s) $name is stopping successful `r")
44      ExitWithCode -ExitCode 0
45  }
46
47  elseif ($action -eq "Start"){
48      Write-host ("$todaydate Start service(s) `r")
49      Get-Service -DisplayName $name | ForEach-Object { if($_.status -eq "Stopped"){Start-Service $name}}
50      Write-host ("$todaydate service(s) $name is starting successful `r")
51      ExitWithCode -ExitCode 0
52  }
53
54  else{
55      Write-Host("Wrong arguments : $action")
56      ExitWithCode -ExitCode 1
57 }
```

RAPPORT DE STAGE SEMAINE 1

APERÇU DES MISSION

Extrait de code que j'ai optimisé
Ce code redémarre le serveur entrée en paramètre

```
param(  
    [string]$Action=$(Throw "Argument Action non renseigné"),  
    [string]$Server=$(Throw "Argument Server non renseigné")  
  
)  
  
2 references  
function Reboot {  
  
    #Redémarre le serveur  
  
    start-sleep -Seconds 10  
    Write-Host "Redémarrage du Serveur $env:ComputerName"  
    Restart-Computer -Force  
    Exit 0  
}
```

Ce code vérifie que le service SQL du serveur existe et si il est stoppé

```
function CheckSqlStop {  
  
    #Vérifie que le service SQL existe et si il est stoppé  
  
    $servicesql = Get-Service -DisplayName "SQL Server (M3)" -ErrorAction SilentlyContinue  
  
    if ($servicesql.name -eq 'MSSQL$M3'){  
        Write-Host "Service $servicesql existant et en status \"$servicesql.status"  
        Stop-Service -Name $servicesql.name -Force  
        Write-Host "Arrêt du service $servicesql en cours"  
  
        Write-Host "service $servicesql en status \"$servicesql.status"  
        if($servicesql.status -eq "Stopped"){  
            Write-Host "Service $servicesql à l'arrêt"  
            Exit 0  
        }  
        else{  
            Write-Host "Erreur lors de l'arrêt du service $servicesql"  
            Exit 1  
        }  
    }  
  
    else{  
        Write-Host "Le service SQL Server (M3) n'existe pas"  
        Exit 0  
    }  
}
```

CONCLUSION

Durant ces premières semaines, j'ai découvert mon environnement de travail au sein de l'entreprise. J'ai appris à utiliser de nouveaux outils comme Wallix et Opcon en manipulant ceux-ci. J'ai également appris à coder dans le langage powershell en travaillant sur les différents scripts.



JANVIER 2024

RAPPORT DE STAGE BTS SIO SLAM

SEMAINE 3

PRÉPARÉ PAR NAM LUAN

CHRISTIAN LOUBOUTIN

DÉCOUVERTE DE L'ENVIRONNEMENT



J'ai découvert les locaux qui sont à Paris dans le 1ère arrondissement à 5 min à pied de châtelet. Les locaux sont spacieux et possèdent plusieurs bureaux équipé d'écran où il suffit d'y brancher son ordinateur portable afin d'avoir accès à la connexion internet, l'écran, un clavier et une souris.



PRÉSENTATION DES NOUVEAUX OUTILS



POSTMAN

Postman API est un outil de développement d'API qui permet aux développeurs de tester, de créer et de partager des API plus efficacement. Il propose une interface utilisateur conviviale pour envoyer des requêtes HTTP, tester des endpoints, organiser des collections d'API, automatiser des tests et générer une documentation interactive. Postman simplifie le processus de développement, de test et de collaboration autour des API, offrant ainsi un moyen pratique de travailler avec les services web.

APERÇU DES **MISSION**

Durant ma 3ème semaines, j'ai continuer mes 2 missions :

- interconnecter via api, l'ordonnanceur de l'entreprise à l'outil Jira
- Mettre en place le stop/start M3 Homologation. Optimiser les scripts existant



APERÇU DES MISSION ET DIFFICULTÉS

Pour l'interconnexion des API j'ai rencontré des difficultés :

Il me manquait des infos pour remplir le scripts permettant la création de ticket Jira.

Pour résoudre ce problème, j'ai d'abord fait un ticket au support Opcon, mais qui au final n'a pas aboutit. J'ai ensuite trouvé une personne au sein de l'entreprise capable de répondre à mes questions. J'ai un call prévu avec lui la semaine prochaine.

Exemple d'un ticket Jira

Create Issue Configure Fields

Project* IT Community (ITCOM)

Issue Type* Incident

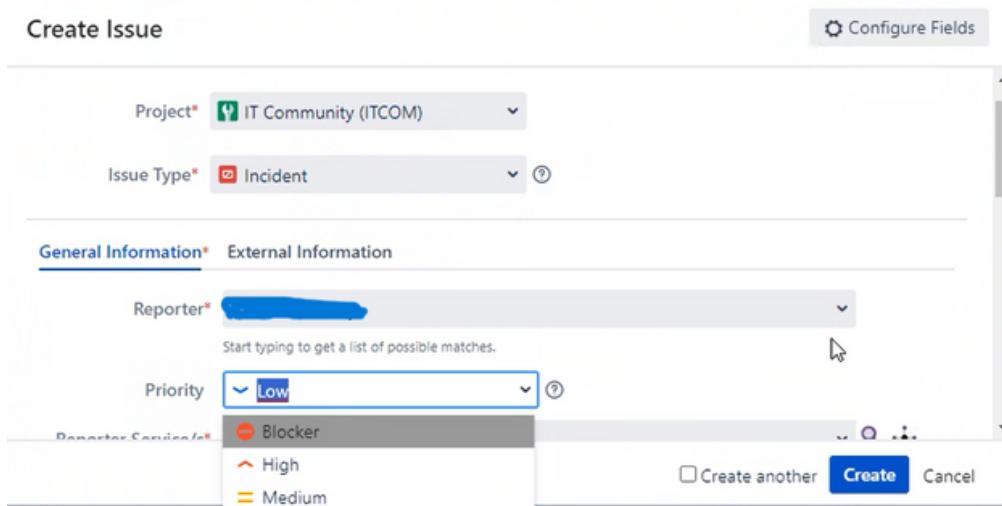
General Information* External Information

Reporter* ██████████ Start typing to get a list of possible matches.

Priority Low

Blocker
High
Medium

Create another Create Cancel



APERÇU DES MISSION ET DIFFICULTÉS

Script permettant la
création du ticket (il me
manquais les infos pour
savoir quel nom de key
ou quel priorité
d'incident il fallait mettre
par exemple)

```
Body (Json)
File Name
Message Body
{
  "fields": {
    "project": {
      "key": "OT"
    },
    "summary": "Job:[[Jl.Uname]] Schedule:[[Jl.Sname]] est en erreur",
    "description": {
      "version": 1,
      "type": "doc",
      "content": [
        {
          "type": "paragraph",
          "content": [
            {
              "type": "text",
              "text": "Le job:[[Jl.Uname]] du Schedule:[[Jl.Sname]] en date du [[SSCHEDULE DATE]] est en erreur avec un code retour = [[Jl.Cret]] sur le serveur : [[Jl.Mach]]."
            }
          ]
        }
      ],
      "issuetype": {
        "name": "Incident"
      },
      "priority": {
        "name": "Medium",
        "id": "3"
      }
    }
  }
}
```

APERÇU DES MISSION ET DIFFICULTÉS

J'ai également continuer mon travail sur les scripts. Durant les semaines passer j'avais déjà fait le gros du travail donc cette semaine-ci je me suis concentrer sur la mise en forme ainsi que des optimisation que je pouvais effectuer

Cette extract de script liste les job en cours d'exécution pour savoir si le Grid peut être stoppé

```
Write-host("`r----- jobs m3 In Progress -----`n")
Write-host ("`r $todayDate Batch jobs -----`n")
foreach ($listbatchjob in $listbatchjobs){
    Write-host ("`n $todayDate Job : " + $listbatchjob.Name +" "+$listbatchjob.owner+" "+$listbatchjob.threadId+" "+$listbatchjob.Status+"`n")
}
Write-host ("`r----- $todayDate Auto jobs -----`n")
foreach ($listautojob in $listautojobs){
    Write-host ("`n $todayDate Job : " + $listautojob.Name +" "+$listautojob.owner+" "+$listautojob.threadId+" "+$listautojob.Status)
}
Write-host ("`n----- $todayDate Interactive jobs -----`n")
foreach ($listinteractivejob in $listinteractivejobs){
    Write-host ("`n $todayDate Job : " + $listinteractivejob.Name +" "+$listinteractivejob.owner+" "+$listinteractivejob.threadId+" "+$listinteractivejob.Status)
}
Write-host ("`n----- $todayDate Mi jobs -----`n")
foreach ($listmijob in $listmijobs)[
    Write-host ("`n $todayDate Job : " + $listmijob.Name +" "+$listmijob.owner+" "+$listmijob.threadId+" "+$listmijob.Status)
]
Write-host("`n----- Jobs Summary -----`r")
Write-host ("$todayDate Batch jobs : $countbatchjobs`r")
Write-host ("$todayDate Auto jobs : $countautojobs`r")
Write-host ("$todayDate Interactive jobs : $countinteractivejobs`r")
Write-host ("$todayDate Mi jobs : $countmijobs`r")

if ($countbatchjobs -eq 0) {
    Write-host ("$todaydate Action : Ready to stop m3 Grid `r")
    Write-host("$todaydate ReturnCode >> 0 `r")
    ExitWithCode -ExitCode 0
}
else {
    Write-host ("$todaydate Action : Warning cannot stop m3 Grid `r")
    Write-host("$todaydate ReturnCode >> 1 `r")
    ExitWithCode -ExitCode 1
}
```

APERÇU DES MISSION ET DIFFICULTÉS

- ▼  Maintenance-M3 (Finished OK / Completed)
 - ▼  Maintenance M3-Redemarrage Grid (Finished OK / Completed)
 - ▼  1-Maintenance-M3-Grid-Stop (Finished OK / Completed)
 -  100-Maintenance-M3-Grid-Jobs-Status (Finished OK - 0000000000)
 -  110-Maintenance-OpCon-AllJobs-Stop (Finished OK - NULL JOB FIN. OK)
 -  115-Maintenance-M3-Asynchrone-Stop (Finished OK - FINISHED_OK)
 -  120-Maintenance-M3-Service-Ion-Stop (Finished OK - 0000000000)
 -  300-Maintenance-M3-Service-ReportManager-Stop_1 (Finished OK - 0000000000)
 -  300-Maintenance-M3-Service-ReportManager-Stop_2 (Finished OK - 0000000000)
 -  300-Maintenance-M3-Service-ReportManager-Stop_3 (Finished OK - 0000000000)
 -  300-Maintenance-M3-Service-ReportManager-Stop_4 (Finished OK - 0000000000)
 -  300-Maintenance-M3-Service-ReportManager-Stop_5 (Finished OK - 0000000000)
 -  310-Maintenance-M3-Service-Streamserve-Stop (Finished OK - 0000000000)
 -  320-Maintenance-M3-Grid-Stop (Finished OK - 0000000000)
 -  325-Timer-GridStop-MEC (Finished OK - 0000000000)
 -  330-Maintenance-M3-Grid-Status (Finished OK - 0000000001)

Exemple de jobs possédant les scripts devant
être modifié.
Dans le logiciel Opcon

JANVIER 2024

RAPPORT DE STAGE BTS SIO SLAM

SEMAINE 4

PRÉPARÉ PAR NAM LUAN

RAPPORT DE STAGE SEMAINE 4

CHRISTIAN LOUBOUTIN



APERÇU DES **MISSION**

Durant ma 4ème semaines, j'ai continué mes 2 missions :

- interconnecter via api, l'ordonnanceur de l'entreprise à l'outil Jira
- Mettre en place le stop/start M3 Homologation. Optimiser les scripts existant



APERÇU DES **MISSION ET DIFFICULTÉS**

Pour l'interconnexion des API j'ai encore rencontré des difficultés :

Après avoir fait mon ticket au support valyantis, une entreprise spécialisée dans la gestion et l'optimisation des plateformes Atlassian, telles que Jira, Confluence et Bitbucket, j'ai reçu une doc qui à pu m'aiguiller.

Pour effectuer une requête API vers Jira, il faut l'url de l'api, et connaître les bon ports ainsi que les logins. Le problème est que les login ne semble pas être bon et je ne suis pas sûr des ports utilisés. Pour régler ces problèmes, je compte faire un ticket au support Jira pour obtenir une assistance sur ce sujet.

APERÇU DES **MISSION ET DIFFICULTÉS**

J'ai beaucoup avancer sur le sujet du Stop/Star M3, en important tout les job depuis le serveur production vers le serveur de staging. Utiliser la machine de staging me permet de tester mes jobs et mes scripts sans que ceux-ci impact les utilisateurs et le bon déroulement des applications.

Il me reste quelque détail à peaufiner, comme la pousser des nouveaux scripts que j'ai écrits sur la machine, en vue d'une présentation sur ce sujet à mon tuteur en fin de semaine.

APERÇU DES

MISSION ET DIFFICULTÉS

Au niveau des principales modifications que j'ai effectués :

J'ai adapter chaque job pour que chacun effectue leurs tâches depuis la bonne machine. Car les jobs importé du serveur de production utilisent des machines en production, on doit donc les changer par des machines d'homologation. La machine de production est l'environnement réel où les utilisateurs finaux accèdent à l'application ou au service, assurant stabilité et performance. En revanche, la machine d'homologation est une copie de l'environnement de production utilisée spécifiquement pour les tests finaux, la validation et l'approbation avant le déploiement en production. L'homologation vise à garantir que les changements logiciels répondent aux exigences et fonctionnent correctement avant d'atteindre les utilisateurs finaux. La machine de production est l'environnement réel où les utilisateurs finaux accèdent à l'application ou au service, assurant stabilité et performance. En revanche, la machine d'homologation est une copie de l'environnement de production utilisée spécifiquement pour les tests finaux, la validation et l'approbation avant le déploiement en production. L'homologation vise à garantir que les changements logiciels répondent aux exigences et fonctionnent correctement avant d'atteindre les utilisateurs finaux.

APERÇU DES

MISSION ET DIFFICULTÉS

Cela permettra de tester chaque job sans impact sur l'environnement de l'entreprise. Je ne peut pas illustrer cette partie pour des raisons de confidentialité.

JANVIER/FÉVRIER 2024

RAPPORT DE STAGE BTS SIO SLAM

SEMAINE 5

PRÉPARÉ PAR NAM LUAN

RAPPORT DE STAGE SEMAINE 5

CHRISTIAN LOUBOUTIN



APERÇU DES **MISSION**

Durant ma 5ème semaines, j'ai continué sur:

- interconnecter via api, l'ordonnanceur de l'entreprise à l'outil Jira
- Mettre en place le stop/start M3 Homologation. Optimiser les scripts existant



APERÇU DES MISSION ET DIFFICULTÉS

Pour l'interconnexion des API :

J'ai fait un ticket pour demander une assistance pour ce projet, car par manque d'info et de documentation sur le sujet, il était compliqué pour moi de le terminer seul.

J'ai eu une réunion avec des personnes de chez valyantis pour qu'ils nous aident à effectuer l'interconnexion API. La personne qui m'a aider m'a montrer la bonne url pour se connecter à l'api ainsi que des modifications à effectuer dans mon script pour que celui-ci affiche les bonnes informations sans encombre. Les tests de requêtes API ont été fait sur le logiciel Postman, il va donc falloir que j'adapte ceux-là pour faire des requêtes depuis l'ordonnanceur Opcon

Script en Json permettant de créer un ticket par API. Chaque ligne correspond à un champ du ticket (la description, le titre, le nom du reporter, la priorité,...)

```
{
  "fields": {
    "project": {
      "id": "10003"
    },
    "summary": "Job: [[JI.Jname]] Schedule: [[JI.Sname]] est en erreur",
    "description": "Le job: [[JI.Jname]] du Schedule: [[JI.Sname]] en date du [[SSCHEDULE_DATE]] est en erreur avec un code retour = [[JI.Cret]] sur le serveur : [[JI.Mach]]",
    "issuetype": {
      "id": "10000"
    },
    "reporter": {
      "name": "NAM"
    },
    "priority": {
      "name": "Medium",
      "id": "3"
    },
    "customfield_11011": [
      {"key": "***"}
    ]
  }
}
```

APERÇU DES

MISSION ET DIFFICULTÉS

Mise en place Stop/Start M3 Homologation :

Durant cette semaine j'ai présenté à mon tuteur de stage les avancés sur ce projet, ce qui me permet d'entrer dans la phase de test des différents jobs et scripts. Cependant il y a quelques détails à corriger, comme vérifier que les liens vers les différents scripts sont les bon sur le serveur d'homologation (Le chemin n'est pas le même que sur le serveur de production), et aussi vérifier que les credentials sont les bon (l'ensemble d'informations permettant d'authentifier l'identité d'un individu ou d'un système). J'ai donc du me connecter au différentes machines pour vérifier les liens et demander les info pour les credentials à une personne ayant celle-ci.

La semaine prochaine, après avoir récupéré toutes les infos nécessaires, je vais lancer la phase de test et de correction des différentes erreurs (si il y en a).

FÉVRIER 2024

RAPPORT DE STAGE BTS SIO SLAM

SEMAINE 6

PRÉPARÉ PAR NAM LUAN

RAPPORT DE STAGE SEMAINE 6

CHRISTIAN LOUBOUTIN



APERÇU DES **MISSION**

Durant ma 6ème semaines, j'ai continué sur:

- interconnecter via api, l'ordonnanceur de l'entreprise à l'outil Jira
- Mettre en place le stop/start M3 Homologation. Optimiser les scripts existant



APERÇU DES MISSION ET DIFFICULTÉS

Pour l'interconnexion des API :

Après plusieurs jour, on a enfin réussi à faire fonctionner la requête API de Opcon vers Jira. Il ne manque plus qu'à améliorer certains détails au niveau du script (ajouter le status du service par ex) et le projet pourra être mit en prod.

Les difficultés était au niveau de l'event (un événement qui se déclenche quand un job tombe en erreur, et qui crée un nouveau job qui, cette fois-ci exécute la requête api vers Jira). Sur cet event, il y avait quelques erreurs de syntaxe qui ne permettaient pas le bon fonctionnement de celui-ci (il manquait un “=”, il y avait des “;” au lieu de “,” à certain endroits).

Event créant un job pour permettre l'appel api

```
[[JOB:ADD [[SSCHEDULE DATE]] Maintenance-Apps_Job-C Alert Status Job[Maintenance-Apps Alert status SCC],API-Issue-SMA-to-JIRA_365j,Iname||[$JOB NAME],Sid||[$SCHEDULE ID],Srn||[$SCHEDULE INST],Cret||[$JOB TERMINATION],Sname||[$SCHEDULE NAME],Mach||[$MACHINE NAME]],
```

APERÇU DES MISSION ET DIFFICULTÉS

Pour l'interconnexion des API :

J'ai continué ma pratique des appels API avec le logiciel Postman.

Après avoir compris l'utilisation des requêtes GET et POST, j'ai voulu essayer de mettre à jour un ticket en modifiant par exemple la description d'un ticket déjà créé.

Pour se faire il suffit d'utiliser une requête PUT et de remplir dans le body le nom du champ que l'on veut modifier ainsi que le contenu souhaité.

J'essaye aussi de comprendre comment ajouter un fichier (un attachements) à un ticket Jira, pour ensuite pouvoir ajouter un fichier de log depuis Opcon lorsqu'un job tombe en erreur. Cela permettra de mieux documenter les tickets.

Requête PUT dans Postman

The screenshot shows the Postman interface with a 'PUT' method selected. The URL field contains 'https://exemple.com/rest/api/2/issue/issueld'. Below the URL, there are tabs for 'Params', 'Authorization', 'Headers (10)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The 'Body' tab is currently active, indicated by an orange underline.

APERÇU DES **MISSION ET DIFFICULTÉS**

Body de la requête PUT (description et attachment sont les noms des champs à modifier)

```
{
  "fields": {
    "description": "Test retour update 2",
    "attachment": [
      {"filename": "test.png"}
    ]
  }
}
```

APERÇU DES **MISSION ET DIFFICULTÉS**

Mise en place Stop/Start M3 Homologation :

J'ai lancé la phase de test des jobs. Lors de la premier exécution de tout les jobs, la plupart étaient en erreurs, ce qui est souvent dû à un problème de credential, de port, ou d'ouverture de flux. Cette semaine j'ai réussi à réparer la plupart des jobs en erreurs. Certains jobs étaient en erreurs dû à une erreur dans mon script, d'autre à cause d'un executable n'étant pas installé sur la machine qui exécute le job, et d'autre à cause d'erreurs au niveaux des adresses ip ou encore des login.

Il me reste encore quelques job en erreurs sur lesquels j'ai des blocages. Je compte les réparer d'ici la semaine prochaine.

APERÇU DES MISSION ET DIFFICULTÉS

Exemple de job (ceux en rouge sont en erreur, ceux en vert sont réussi)

- ✓  Maintenance M3-HOMO-Patch Management (Failed / Completed)
- ✓  100 -HOMO-Stop Grid (Failed / Completed)
 -  100-Maintenance-M3-Grid-Jobs-Status (Failed - 0000000001)
 -  110-Maintenance-OpCon-AllJobs-Stop (Finished OK - NULL JOB FIN. OK)
 -  115-Maintenance-M3-Asynchrone-Stop (Failed - 31007)
 -  120-Maintenance-M3-Service-Ion-Stop (Finished OK - 0000000000)
 -  140-Maintenance-M3-Service-ReportManager-Stop_1 (Finished OK - 0000000000)
 -  140-Maintenance-M3-Service-ReportManager-Stop_2 (Finished OK - 0000000000)
 -  140-Maintenance-M3-Service-ReportManager-Stop_3 (Finished OK - 0000000000)
 -  140-Maintenance-M3-Service-ReportManager-Stop_4 (Finished OK - 0000000000)
 -  140-Maintenance-M3-Service-ReportManager-Stop_5 (Finished OK - 0000000000)
 -  150-Maintenance-M3-Service-Streamserve-Stop (Finished OK - 0000000000)
 -  160-Maintenance-M3-Grid-Stop (Finished OK - 0000000000)
 -  170-Timer-GridStop-MEC (Finished OK - 0000000000)

FÉVRIER 2024

RAPPORT DE STAGE BTS SIO SLAM

SEMAINE 7

PRÉPARÉ PAR NAM LUAN

RAPPORT DE STAGE SEMAINE 7

CHRISTIAN LOUBOUTIN



APERÇU DES **MISSION**

Durant ma 7ème semaines, j'ai continué sur:

- interconnecter via api, l'ordonnanceur de l'entreprise à l'outil Jira
- Mettre en place le stop/start M3 Homologation. Optimiser les scripts existant



APERÇU DES **MISSION ET DIFFICULTÉS**

Pour l'interconnexion des API :

Le job ainsi que son script ont fini d'être développé. Dès qu'un job tombe en erreur, un job est créé et il récupère le nom du job, le nom de son schedule, la date du jour, le code de retour et le nom du serveur puis crée un ticket sur la plateforme Jira par requête API.

Pour améliorer l'automatisation de la création des tickets, il faut rajouter automatiquement les reporter services (les noms des services dont appartient le job en erreur), car sans ça, il est obligé de les ajouter à la mains. Pour ce faire il faudrait parcourir la string contenant le nom du schedule du job en erreur et la faire correspondre avec les noms des reporters services disponible. Une fois que le nom du ou des reporters services est trouvé, il suffit juste de l'ajouter au ticket directement dans la requête en POST ou après en effectuant une requête PUT.

RAPPORT DE STAGE SEMAINE 7

APERÇU DES MISSION ET DIFFICULTÉS

Ticket Jira avec un reporter service (Opcon)

 Test retour update

[Edit](#) [Add comment](#) [Assign](#) [More](#) [Escalate Stream](#) [Escalate Region](#) [Wor](#)

Details

Type:	<input checked="" type="checkbox"/> Incident	Status:	OPEN
Priority:	<input checked="" type="checkbox"/> Medium	(View Workflow)	
		Resolution:	Unresolved

[General Information](#) [Escalation](#) [External Information](#)

Reporter Service/s: [Opcon](#)

Service Status: Degraded

Description

Test retour update 2

APERÇU DES **MISSION ET DIFFICULTÉS**

Le script sur lequel se base ma requête est en Json , qui est un langage ne permettant pas d'effectuer des conditions tels que "if" ou "else et également des boucles comme "for". Donc j'ai essayé de développer un script en javascript mais, j'utilise Opcon pour faire mes appels API, et le module permettant de faire ces requêtes ne prends que des scripts en Json.

L'idée serait de créer un autre job qui traitera les données et retournera une ou plusieurs valeurs de reporter services , pour ensuite lancer un autre job qui fera une requête PUT pour insérer le reporter services dans le ticket Jira.

APERÇU DES **MISSION ET DIFFICULTÉS**

Mise en place Stop/Start M3 Homologation :

Au niveau des infos manquantes, le support M3 m'a répondu mais je n'ai pas accès à la ressource où les infos sont disponibles. Je dois donc demander à quelqu'un qui a les accès pour obtenir les informations.

J'ai également fait un ticket pour la demande de création d'un compte nécessaire au fonctionnement d'un job dans l'ordonnanceur Opcon.

Le reste des jobs ne sont pas en erreur donc il ne me reste plus que les deux cités au-dessus pour pouvoir commencer la migration vers la prod.

FÉVRIER 2024

RAPPORT DE STAGE BTS SIO SLAM

SEMAINE 8

PRÉPARÉ PAR NAM LUAN

RAPPORT DE STAGE SEMAINE 8

CHRISTIAN LOUBOUTIN



APERÇU DES **MISSION**

Durant ma 8ème semaines, j'ai continué sur:

- interconnecter via api, l'ordonnanceur de l'entreprise à l'outil Jira
- Mettre en place le stop/start M3 Homologation. Optimiser les scripts existant



APERÇU DES **MISSION ET DIFFICULTÉS**

Pour l'interconnexion des API :

Au final, automatiser l'ajout de reporter service sur Jira en se basant sur les jobs de l'ordonnanceur est trop compliqué car le langage Json ne permet pas d'utiliser des conditions comme "if", ou "else". Il serait possible cependant d'utiliser un job par lequel sera traité les données du ticket et un autre qui fera une requête pour importer les données traitées vers le ticket Jira mais cela reste assez complexe.

J'ai également demandé la création d'un compte dédié à la création de ticket dans Jira mais cela s'avérait compliqué à cause des licences qui sont limitées.

APERÇU DES **MISSION ET DIFFICULTÉS**

Mise en place Stop/Start M3 Homologation :

Pour les besoins d'un job, j'ai fait un call avec des personnes travaillant chez infor qui m'ont crée les keystore name et password (des logins permettant l'exécution du script), ce qui m'a permis d'avancer sur ce projet. J'ai dû parler en anglais avec eux car ils n'étaient pas français, ce qui a aussi permis de renforcer mon niveau de langue !

J'ai créé à l'aide d'un collègue un ticket pour demander la création d'un compte opcon dans l'environnement M3 homologation pour l'exécution de certains scripts.

Sur d'autres scripts j'ai optimisé le fichier de sortie en améliorant notamment la mise en page du fichier, pour le rendre plus lisible. A l'aide des `n (saut de ligne) et `r (retour à la ligne).

APERÇU DES MISSION ET DIFFICULTÉS

Exemple sur le code

```
Write-host ("`r")
Write-host("`r`n----- Jobs Summary -----")
# Affiche le nombre de chaque jobs
Write-host ("`r`n $todayDate Batch jobs : $countbatchjobs")
Write-host ("`r`n $todayDate Auto jobs : $countautojobs")
Write-host ("`r`n $todayDate Interactive jobs : $countinteractivejobs")
Write-host ("`r`n $todayDate Mi jobs : $countmijobs")
```

Exemple dans le code de retour

```
----- Jobs Summary -----
2024-02-22-092540485 Batch jobs : 0
2024-02-22-092540485 Auto jobs : 81
2024-02-22-092540485 Interactive jobs : 0
2024-02-22-092540485 Mi jobs : 24
```

APERÇU DES MISSION ET DIFFICULTÉS

Après vérification les login que m'ont donnée le support d'Infor sont bon mais n'ont pas assez de privilège. J'ai cette erreur en sortie de mon job :

```
| insufficient credentials - supplied certificate is not granted the grid-admin role
```

Il faudra donc refaire un ticket ou leur demander dans le même ticket de régler ça.

Une fois que ce job ne sera plus en erreur toute la partie de staging sera terminer et on pourra entamer la mise en production du projet.

J'ai optimiser un script en supprimant une condition qui n'était pas logique dedans (voir la condition en commentaire "#")

```
try
{
    $javacmd="java -cp " + $jardir+"\\"+"grid-core-"+$gridVersion+".jar "+$gridClass+ " "+$gridServer+" "+$gridPort
    invoke-expression $javacmd | out-string -outvariable $result
    $isOnline = $result -like "*ONLINE grid is online*"

    if ($isOnline) {
        Write-Host "`n$todayDate Action $gridCommand executed successfully, Grid is Online"
        ExitWithCode -ExitCode 0
    }
    # else {
    #     Write-Host "`n$todayDate Action $gridCommand executed successfully, Grid is Offline"
    #     ExitWithCode -ExitCode 1
    # } ?
}

catch
{
    Write-host ("`n $todayDate Action $gridCommand error`r")
    ExitWithCode -ExitCode 1
}
```

APERÇU DES MISSION ET DIFFICULTÉS

J'ai manipulé une autre fonctionnalité d'opcon qui permet de décider en fonction du code de sortie si le job en question sera en "Failed" ou en "Finish OK" (réussite).

Advanced

Exit Codes					
And/Or	Comparison Operator	Value	End Value	=	Result
⊕	Not Equal To	0	0	=	Finish OK
⊗		0	0	=	Finish OK
⊗		0	0	=	Finish OK
⊗		0	0	=	Finish OK

Anything Else
Fail

Ici, si le code de sortie n'est pas égal à 0 (donc, la plupart du temps, s'il est égal à 1), le job sera en "Finish OK". Si le code de sortie est différent de 0, il sera en "Failed".

Ces changements me sont utiles pour un job qui vérifie le status d'un service, s'il est lancé le code de sortie sera 0 s'il ne l'est pas il sera de 1.

APERÇU DES MISSION ET DIFFICULTÉS

J'ai améliorer mes compétences powershell au niveaux du rendu visuel des retour, avec de l'ajout de couleurs au niveau des textes de retour dans la console.

résultat en sortie

```
Action executed successfully, Grid is Online  
Action error
```

code

```
1 Write-Host "`n$todayDate Action $gridCommand executed successfully, Grid is Online" -ForegroundColor Green  
2 Write-host "`n $todayDate Action $gridCommand error`r" -ForegroundColor Red
```

A l'aide de “-ForegroundColor” qui permet de changer la couleur des textes de retour dans la console. Cela peut permettre par exemple de faciliter la compréhension du fichier de retour (Vert, il fonctionne / Rouge il est en erreur).

FÉVRIER 2024

RAPPORT DE STAGE BTS SIO SLAM

SEMAINE 9

PRÉPARÉ PAR NAM LUAN

RAPPORT DE STAGE SEMAINE 9

CHRISTIAN LOUBOUTIN



APERÇU DES **MISSION**

Durant ma 8ème semaines, j'ai continué sur:

- interconnecter via api, l'ordonnanceur de l'entreprise à l'outil Jira
- Mettre en place le stop/start M3 Homologation. Optimiser les scripts existant



APERÇU DES MISSION ET DIFFICULTÉS

Pour l'interconnexion des API :

Mise en place sur les job Y2 pour vérifier si cela fonctionne bien.

Le lendemain, j'ai vérifier et le job qui crée les tickets est bien fonctionnel.

Il y a bien des tickets qui ont été crée sur Jira pour les jobs qui sont tombé en erreurs dans la soirée, et ils ont bien été placé dans le bon répertoire.

J'ai donc déployer mon job sur le reste des job ciblé (M3 et EPM).

Exemple de ticket crée par mon script dans Jira lorsqu'un job est en ereurs

OPCON ERREUR JOB									Save as	Details	★	Share	Export	Tools
IT Community		Incident	Open	Assignee: All	Contains text	More	Search	Advanced						
Reporter Service/s:		Reporter: CDS Jira ServiceNav									Columns	☰		
T	Key	Summary			Assignee	Reporter	P	Status	Created	Resolved	Reporter Service/s	Change Origin		
1	[REDACTED]	Job:100-[REDACTED] Schedule [REDACTED]	est en erreur			Unassigned	CDS Jira ServiceNav	=	OPEN	27/02/24		Opcon		
2	[REDACTED]	Job:100-[REDACTED] Schedule [REDACTED]	est en erreur			Unassigned	CDS Jira ServiceNav	=	OPEN	27/02/24		Opcon		

APERÇU DES **MISSION ET DIFFICULTÉS**

Pour l'interconnexion des API :

Le derniers jours après vérification du bon fonctionnement des jobs de création de tickets, nous avons appliqué cela pour tout les jobs disponible.

APERÇU DES

MISSION ET DIFFICULTÉS

J'ai animé un meeting avec l'équipe IT dans le cadre d'un "operations update" : Meeting où l'on donne notre avancement sur les projets annoncés en début de semaine.

APERÇU DES **MISSION ET DIFFICULTÉS**

Mise en place Stop/Start M3 Homologation :

Après avoir fait un call avec le support Infor (encore en anglais, j'ai obtenu les bon credentials pour mon job. J'ai pu tester et voir que mon job fonctionnait bien sur le serveur de staging et sur le serveur de prod également.

Cela signifie donc que tout les jobs sont fonctionnel et qu'ont peut commencer à faire les tests sur la prod.

J'ai donc effectuer une migration à l'aide d'un collègue pour transférer tout les jobs du serveur de staging vers le serveur de prod. J'ai ensuite dû tout reconfigurer dans la prod car il faut changer certains nom de machine d'homologation par celles de productions. Il a fallu faire de même pour les scripts qui eux, n'ont pas été importé. Il faut donc mettre à jour les scripts dans le repository et les configurer sur les jobs. Pour certains job il a fallu également vérifier les ports utilisés par la machine de destination.

APERÇU DES **MISSION ET DIFFICULTÉS**

Mise en place Stop/Start M3 Homologation :

Lors de mon dernier jour de stage, avec des membres de mon équipes, nous avons effectué des test sur la prod actuelles (pour ensuite mettre tout les jobs en prod si ils fonctionnent tous).

J'ai donc lancer les jobs et j'ai corrigé ceux qui ont finit en erreurs au fur et à mesure de l'exécution. A la fin du test la plupart des jobs ont fini sans erreurs (suite au correction si nécessaire).